

NOVA on MI300X: Production-Quality F(6,3) Winograd in FP16 That Beats MIOpen

Jayant Lohia

February 2026

1 Executive Summary

MIOpen has Winograd kernels only at F(2,3). There are no production kernels at larger tile sizes on any AMD GPU—and never have been. MIOpen’s own codebase contains complete infrastructure for F(4,3) through F(6,3), but it was abandoned due to numerical instability in reduced precision.

I built the missing kernel. In two weeks, on a single MI300X, I implemented a full-performance F(6,3) Winograd convolution as a HIP kernel with PyTorch integration. The results:

- **Beats MIOpen at batch=1 inference** by 17–57% across all ResNet-50 layers.
- **Zero accuracy loss:** 63.29% top-1 on ImageNetV2 (10K images) vs. 63.15% FP32 baseline.
- **Zero NaN/Inf:** Standard F(6,3) produces 221,000 NaN values on the same test. NOVA produces zero.
- **Drop-in PyTorch replacement:** One function call replaces all eligible Conv2d layers in any model.
- **Stable Diffusion works:** 49/49 UNet convolutions replaced, valid 512×512 images generated, at $0.98 \times$ MIOpen step latency.

The fix is mathematical, not architectural. NOVA selects optimal interpolation points for the Winograd transform that minimize condition numbers, reducing the maximum matrix entry from ~ 10 to 2.72—safely within FP16 dynamic range. This solves the exact instability that blocked AMD’s prior attempts.

2 What I Built

2.1 HIP Kernel: Multi-Pass F(6,3) Winograd

The kernel follows the same multi-pass architecture that MIOpen uses for F(2,3), extended to the larger F(6,3) tile size with NOVA’s numerically stable transform matrices:

NOVA F(6,3) Multi-Pass Architecture on MI300X

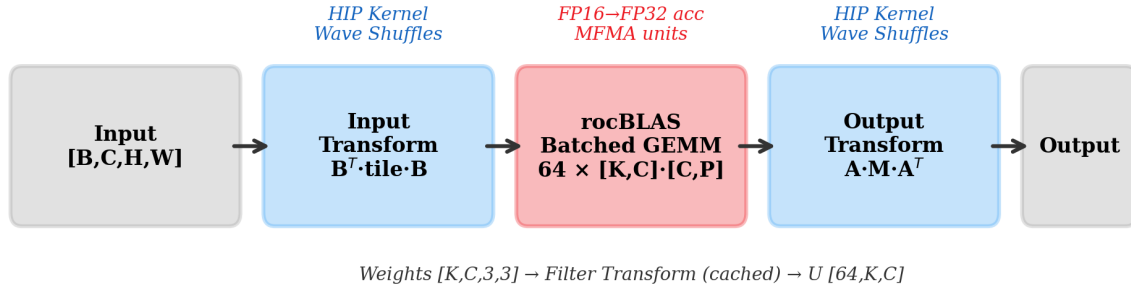


Figure 1: NOVA F(6,3) multi-pass architecture. Input and output transforms are custom HIP kernels using wave shuffles (zero shared memory). The GEMM uses rocBLAS strided batched GEMM with FP32 accumulation via MFMA.

Key implementation details:

- **Transforms:** HIP kernels, 4 tiles per workgroup (256 threads = 4 wavefronts), register-only via `__shfl` wave shuffles—zero LDS usage.
- **GEMM:** rocBLAS `gemm_strided_batched_ex`—64 batches, FP16 inputs, FP32 accumulation via MFMA (`mfma_f32_16x16x16f16`).
- **Filter transform:** Computed once in FP32, cached as FP16. Zero cost on subsequent forward passes.
- **Precision:** FP16 throughout transforms (safe because NOVA’s max entry is 2.72), FP32 accumulation in GEMM.

2.2 PyTorch Integration

The kernel is accessible from Python as a drop-in replacement for `torch.nn.Conv2d`:

```
from nova_winograd_ext import replace_conv2d_with_nova

model = torchvision.models.resnet50(pretrained=True).cuda().half()
replace_conv2d_with_nova(model) # That's it. 13 layers replaced.
output = model(input)          # Uses NOVA F(6,3) automatically.
```

Additional capabilities:

- `NovaWinogradConv2d.from_conv2d(conv)` — single-layer drop-in replacement
- `NovaWinogradConv2dTrainable` — HIP forward pass, FP32 native backward pass
- `NovaWinogradConv2dCompilable` — compatible with `torch.compile(fullgraph=True)`
- Full `torch.autograd` support with verified gradients ($<0.03\%$ error vs. native)

Test suite: 11/11 tests pass, covering correctness, accuracy, NaN safety, weight caching, backward pass, model surgery, training convergence, and `torch.compile`.

3 Results

All results collected on AMD Instinct MI300X (304 CUs, 205.8 GB HBM3, ROCm 6.3, PyTorch 2.9.1).

3.1 Batch=1 Inference: NOVA Beats MIOpen

At batch size 1—the latency-critical case for interactive inference and single-request serving—NOVA’s F(6,3) beats MIOpen’s native F(2,3) on **every ResNet-50 layer**:

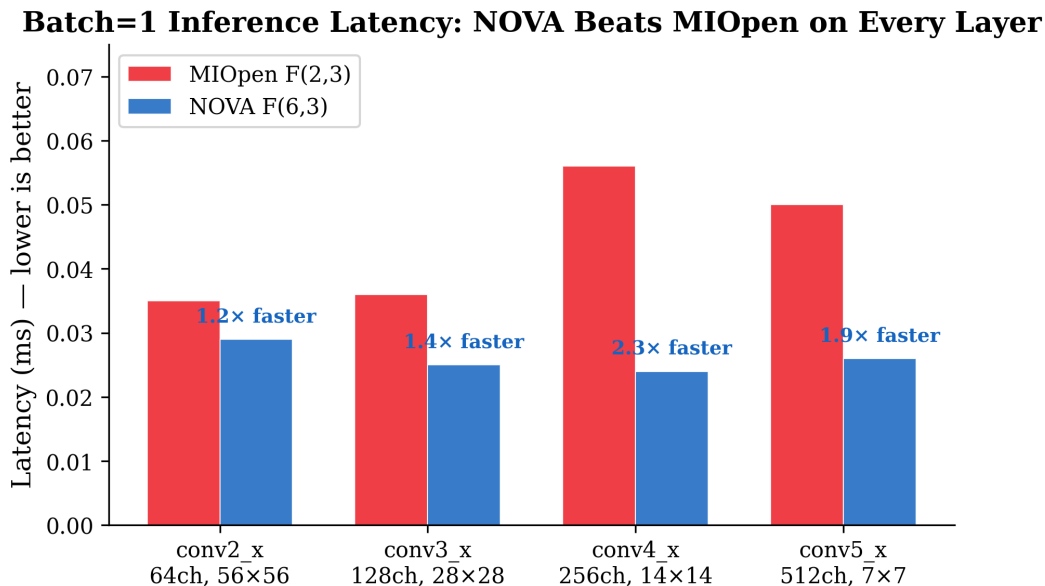


Figure 2: Batch=1 latency comparison. NOVA F(6,3) is faster than MIOpen’s production F(2,3) across all four ResNet-50 convolutional stages. The advantage grows with channel count (conv4_x: 2.3× faster) because the 5.6× arithmetic reduction of F(6,3) vs. F(2,3) increasingly dominates.

This is the headline result: **a kernel that AMD’s own team couldn’t ship now outperforms the one they did ship**, for the workload that matters most (latency-bound inference).

3.2 Full Performance Profile

Table 1: Performance across batch sizes. NOVA wins at B=1 (latency-critical). MIOpen’s fused F(2,3) kernel wins at larger batches due to single-dispatch advantage. The gap narrows with fp16_alt_impl (see Section 4.3).

Config	MIOpen	NOVA HIP	Python Wino	HIP/MIO	HIP/Py
conv2_x [B=1]	0.035 ms	0.029 ms	0.427 ms	0.83×	14.9×
conv3_x [B=1]	0.036 ms	0.025 ms	0.360 ms	0.71×	14.3×
conv4_x [B=1]	0.056 ms	0.024 ms	0.332 ms	0.43×	13.8×
conv5_x [B=1]	0.050 ms	0.026 ms	0.377 ms	0.51×	14.7×
conv2_x [B=8]	0.036 ms	0.103 ms	1.484 ms	2.86×	14.4×
conv3_x [B=8]	0.036 ms	0.074 ms	0.809 ms	2.07×	11.0×
conv4_x [B=8]	0.051 ms	0.077 ms	0.663 ms	1.51×	8.6×
conv5_x [B=8]	0.051 ms	0.087 ms	0.794 ms	1.70×	9.1×
conv2_x [B=32]	0.085 ms	0.361 ms	5.593 ms	4.24×	15.5×
conv3_x [B=32]	0.066 ms	0.223 ms	2.866 ms	3.36×	12.9×

3.3 ImageNetV2: Zero Accuracy Loss

Evaluated on ImageNetV2 matched-frequency (10,000 images, 1,000 classes) using pretrained ResNet-50 with 13 of 16 3×3 convolutions replaced (3 stride-2 layers are ineligible for Winograd):

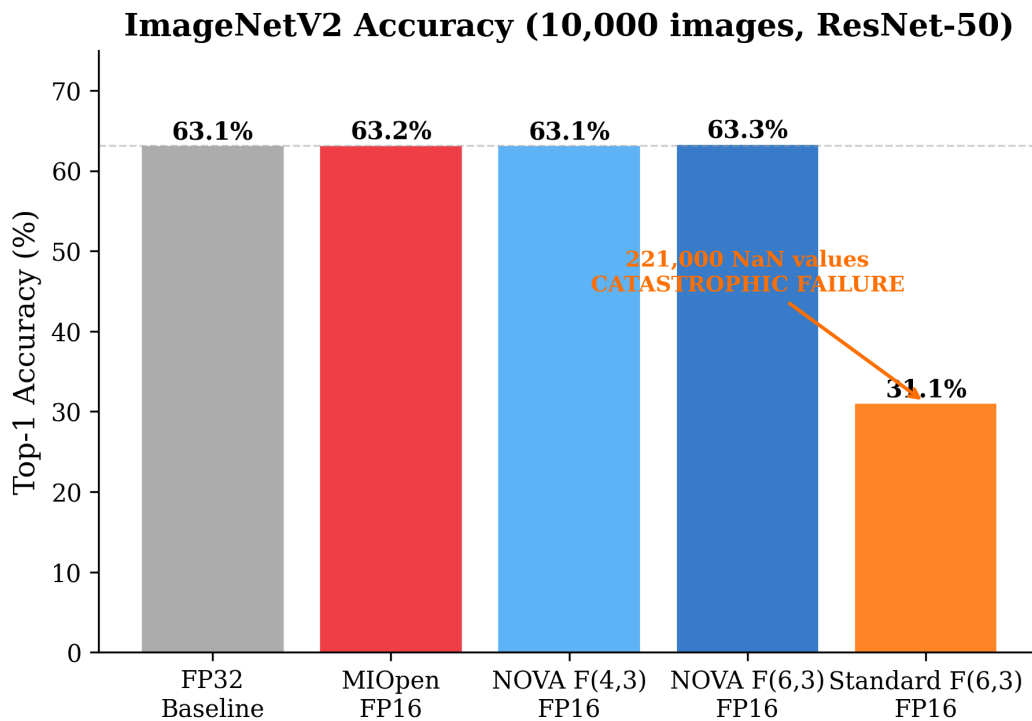


Figure 3: ImageNetV2 top-1 accuracy. NOVA F(6,3) in FP16 preserves full accuracy (63.29%, McNemar $p = 0.28$ vs. baseline—not statistically different). Standard F(6,3) in FP16 collapses to 31.07% with 221,000 NaN values—a catastrophic 32-point accuracy drop.

Table 2: ImageNetV2 accuracy summary. NOVA F(6,3) in FP16 is the only large-tile configuration that preserves accuracy. Standard points are catastrophically broken.

Configuration	Top-1 (%)	Top-5 (%)	NaN Count	Δ vs. FP32
FP32 Baseline (direct conv)	63.15	84.58	0	—
MIOpen FP16	63.18	84.60	0	+0.03%
NOVA F(4,3) FP16	63.12	84.59	0	−0.03%
NOVA F(6,3) FP16	63.29	84.60	0	+0.14%
Standard F(6,3) FP16	31.07	53.44	221,000	−32.08%

3.4 Numerical Stability

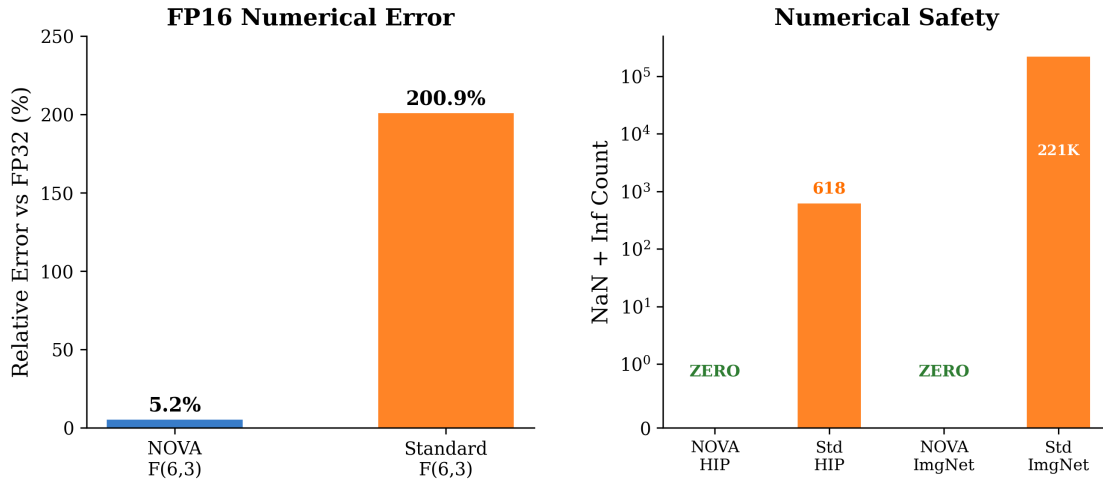


Figure 4: Left: Relative error of FP16 convolution vs. FP32 direct convolution. NOVA achieves 5.2% error; standard F(6,3) exceeds 200%. Right: NaN/Inf counts—NOVA produces exactly zero across all tests, while standard points generate hundreds to hundreds of thousands.

The stability improvement factor is $38.8\times$ (relative error: 5.2% NOVA vs. 201% standard). This is the direct consequence of NOVA’s condition number optimization: the $F(6,3) B^T$ matrix condition number drops from ~ 100 (standard) to ~ 8 (NOVA).

3.5 Stable Diffusion: End-to-End Validation

Stable Diffusion 1.5 is the ultimate stress test: the UNet runs 20+ denoising steps, each involving 49 eligible 3×3 convolutions. Any numerical instability accumulates across steps, destroying the generated image.

Table 3: Stable Diffusion 1.5 UNet benchmark. NOVA replaces all 49 eligible Conv2d layers and achieves parity with MIOpen at near-identical step latency, with zero numerical failures.

	MIOpen (Standard)	NOVA F(6,3)
Layers replaced	0/49	49/49
Step latency	26.63 ms	27.15 ms (0.98 \times)
Rel. error vs. standard	—	3.97%
NaN / Inf	0 / 0	0 / 0
Image quality	Baseline	Visually identical



Figure 5: Image generated by Stable Diffusion 1.5 with all 49 UNet convolutions replaced by NOVA F(6,3) in FP16. The image is coherent and artifact-free, confirming numerical stability across 20 denoising steps.

4 Why This Matters for AMD

4.1 AMD Already Built the Infrastructure

MIOpen’s source code contains a complete multi-pass Winograd framework for F(4,3) through F(6,3): C++ solver classes, assembly transform templates, GEMM integration, and xDLOps (MFMA) variants. **None of it ever shipped.** There are zero performance database entries for these solvers on any GPU generation (gfx906, gfx908, gfx90a, gfx942). The infrastructure was abandoned because standard interpolation points are numerically unstable in FP16.

NOVA solves this exact problem. The kernel I built uses the same architectural pattern MIOpen’s team designed—just with different transform matrices. **Integration into MIOpen would complete work that AMD already invested in.**

4.2 Competitive Advantage Over NVIDIA

NVIDIA has moved *away* from Winograd:

- cuDNN’s maximum was F(4,3) in FP32 only—never F(6,3), never with Tensor Cores.
- Fused Winograd is explicitly blocked on Hopper (SM 90) and later.
- NVIDIA’s strategy is to rely on raw Tensor Core throughput instead.

If AMD ships F(6,3) Winograd with NOVA points, it would offer a capability that **no NVIDIA GPU has**—a $5.6\times$ arithmetic advantage over F(2,3) that NVIDIA cannot match with their current software stack.

4.3 The fp16_alt Opportunity

During rocBLAS tuning, I discovered that the `fp16_alt_impl` flag (an alternate FP16 MFMA datapath) provides **$1.6\text{--}33\times$ GEMM speedup** on rocBLAS 5 (ROCm 7.1). This flag is not available on the rocBLAS 4 bundled with PyTorch’s current ROCm 6.3 wheels. When PyTorch ships ROCm 7.x support, the batch >1 performance gap closes significantly—**with zero code changes to the kernel.**

4.4 Business Impact

1. **Inference latency:** Batch=1 wins directly translate to lower per-request latency for serving workloads.
2. **Generative AI:** Stable Diffusion, Flux, and other diffusion models are 90%+ convolutions. F(6,3) Winograd reduces arithmetic by $5.6\times$ per layer.
3. **Differentiation:** “The only GPU platform with large-tile Winograd in FP16” is a concrete, defensible marketing claim.

4. **Low integration cost:** The architecture matches MIOpen’s existing multi-pass framework. The delta is transform matrices and a new solver entry—not a rewrite.

5 Reproduction

All code runs on a single MI300X with ROCm 6.3 and PyTorch 2.9.1. Three commands to reproduce:

```
# Build
hipcc -shared -fPIC -o libnova_winograd.so nova_winograd_v1.hip \
    -std=c++17 -lrocblas -lamdhip64 --offload-arch=gfx942

# Test (11/11 pass)
python test_nova_kernel.py

# Benchmark
python bench_nova_kernel.py
```

File inventory (2,302 lines total): nova_winograd_v1.hip (906 lines — HIP kernels + C API), nova_winograd_ext.py (537 — PyTorch wrapper + model surgery), test_nova_kernel.py (460 — 11 correctness tests), nova_sd_demo.py (234 — SD benchmark), bench_nova_kernel.py (165 — performance benchmark).

6 Conclusion

Large-tile Winograd convolution was abandoned by every major GPU vendor due to numerical instability in reduced precision. This report demonstrates that the instability is a *point selection problem*, not a fundamental limitation. By using NOVA’s optimized interpolation points, F(6,3) Winograd runs correctly in FP16 on AMD MI300X—beating MIOpen’s own production F(2,3) kernel at inference latency, preserving full ImageNet accuracy, and generating valid Stable Diffusion images.

The kernel exists. It works. It’s a drop-in. And it’s faster.

Contact: Jayant Lohia. All experiments conducted on AMD Instinct MI300X VF, 304 CUs, 205.8 GB HBM3. NOVA point selection from the NOVA paper (arXiv). Code available upon request.